



**ADVANCED**  
NETWORK DEVICES

## CW Outbound Rest Requests For Chat Services

Version 1.1

3/30/2026

2025 ADVANCED NETWORK DEVICES  
3820 NORTH VENTURA DR.  
ARLINGTON HEIGHTS, IL 60004  
U.S.A

ALL RIGHTS RESERVED

## PROPRIETARY NOTICE AND LIABILITY DISCLAIMER

The information disclosed in this document, including all designs and related materials, is the valuable property of Digital Advanced Network Devices and/or its licensors. Advanced Network Devices and/or its licensors, as appropriate, reserve all patent, copyright and other proprietary rights to this document, including all design, manufacturing, reproduction, use, and sales rights thereto, except to the extent said rights are expressly granted to others.

The Advanced Network Devices product(s) discussed in this document are warranted in accordance with the terms of the Warranty Statement accompanying each product. However, actual performance of each product is dependent upon factors such as system configuration, customer data, and operator control. Since implementation by customers of each product may vary, the suitability of specific product configurations and applications must be determined by the customer and is not warranted by Advanced Network Devices.

To allow for design and specification improvements, the information in this document is subject to change at any time, without notice. Reproduction of this document or portions thereof without prior written approval of Advanced Network Devices is prohibited.

### Static Electric Warning



## TROUBLESHOOTING AND ADDITIONAL RESOURCES

User Support: <https://www.anet.com/user-support/>  
Technical Resources: <https://www.anetd.com/user-support/technical-resources/>  
AND Legal Disclaimer: <https://www.anetd.com/legal/>

## OVERVIEW

ClockWise Campus 1.17.27 introduced Outbound REST Requests in CW Web. This enables users to send ClockWise messages to various chat services including Google, MS Teams, Webex, Zoom, and Nextcloud.

Although these are the only services that were tested during development of this feature, it is likely that ClockWise would be able to send messages through additional third-party services that support webhooks. If you plan to use a service not listed here, please refer to that service's documentation for webhook setup instructions, as most webhook-enabled services provide publicly available guidance.

There are three major steps that need to be performed to accomplish this. You will first need to set up the webhook for the chat service. Next, use the webhook information to generate the API endpoint in Clockwise Web. Finally, the API endpoint can be selected as a recipient for any ClockWise notification.

## SETTING UP THE WEBHOOK

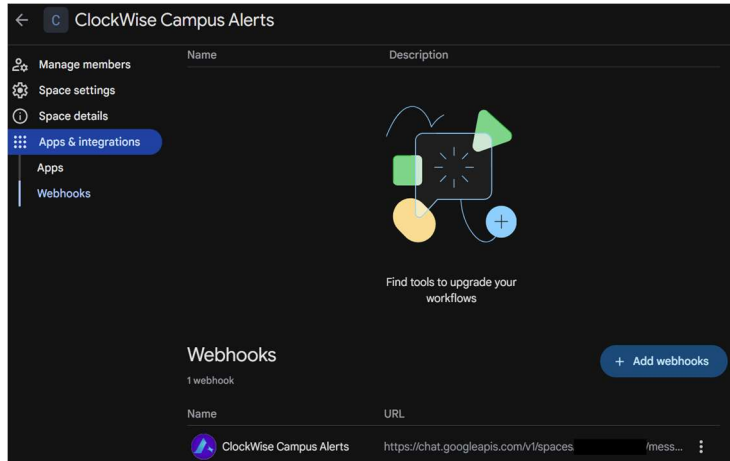
Each service has their own method for setting up the webhook. The various services that have been tested during the development of this feature are included in this document.

Additional information on setting up the webhook for these different services can be found below. Some services require slightly different configurations. These differences will be pointed out in their specific section.

You will need permissions to generate the webhook and add it to the chat or channel.

## SETTING UP WEBHOOKS FOR GOOGLE CHAT

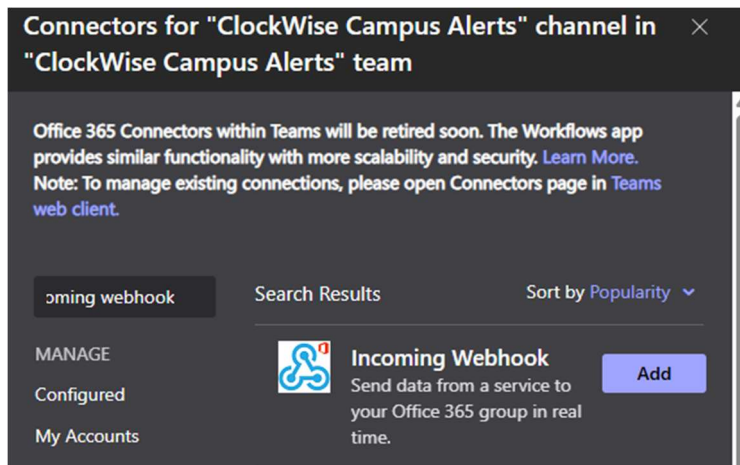
You can add a webhook to any Google Chat or space in the Apps & integrations section for the chat:



Click the Add webhooks button and give it a name and an image URL (optional). You can retrieve the URL for the webhook by clicking the three dots and selecting the copy link button.

## SETTING UP WEBHOOKS FOR MS TEAMS

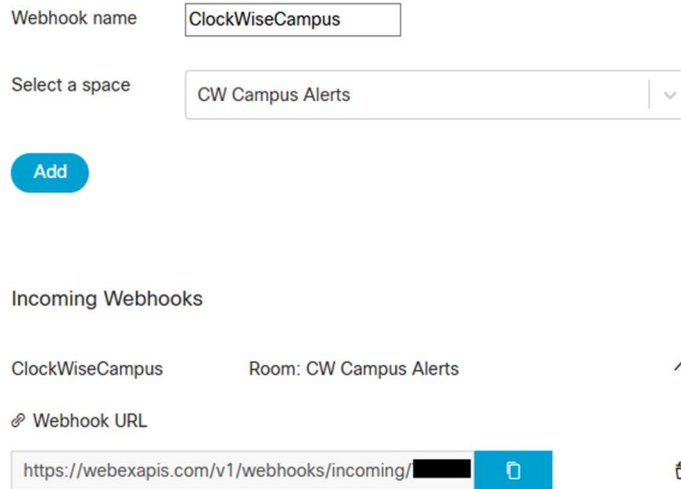
A webhook can be created for any MS Teams channel. First, create or find a pre-existing channel. Once the channel is created, click the three dots next to the name and choose Manage Channel. From there, click the edit button in the Connectors section and search for "Incoming Webhook":



Click the Add button and configure the webhook with a name and image of your choosing. Once created, copy the URL and save it for the next step.

## SETTING UP WEBHOOKS FOR WEBEX

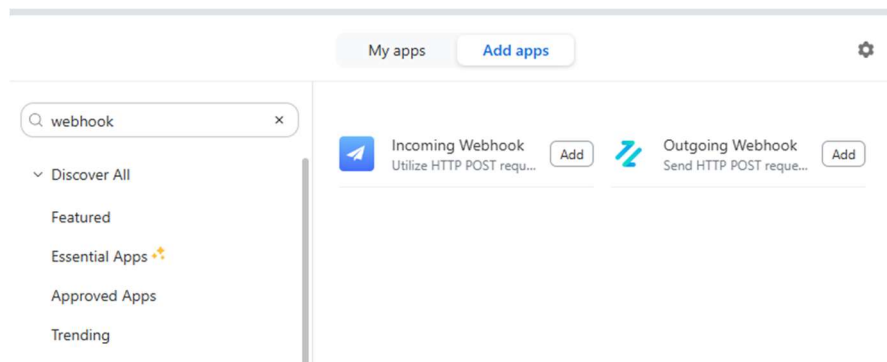
Connect the [Incoming Webhooks](#) app in the Webex App Hub. The webhook can be added to the space at the bottom of the Incoming Webhooks page. Enter a name, then Select a space, and click the **Add** button. Once added, a Webhook URL will be generated automatically:



The screenshot shows the 'Incoming Webhooks' configuration page in Webex. At the top, there is a form with two fields: 'Webhook name' containing 'ClockWiseCampus' and 'Select a space' with a dropdown menu showing 'CW Campus Alerts'. Below the form is a blue 'Add' button. Underneath, the 'Incoming Webhooks' section displays the added webhook: 'ClockWiseCampus' in the 'Room: CW Campus Alerts'. A 'Webhook URL' field is shown with a partially redacted URL: 'https://webexapis.com/v1/webhooks/incoming/[redacted]'. There are copy and delete icons next to the URL.

## SETTING UP WEBHOOKS FOR ZOOM

In the Zoom desktop app, click the apps button and search for “webhook”. Add the “incoming Webhook” to your account:



The app should be visible in the Apps section on the Team Chats page. Now that the app has been added, you can send “/inc connect Incoming Webhook” in the chat or channel to add the incoming webhook app. The webhook will give you the necessary URL and additional information for the next step.

You must manually add a header in the following step if using Zoom. The Header **Key** is “Authorization,” and the **Value** will be the Verification Token that the webhook provides in the chat or channel.

Headers		
Key	Value	Actions
<input type="text" value="Authorization"/>	<input type="text" value="hQJR6AX2R3qLu5Z5y7IrmQ"/>	

## SETTING UP WEBHOOKS FOR NEXTCLOUD

To add a webhook in Nextcloud, you will first need to create a bot account and add it to the conversation that you would like to send automated chats to.

Unlike the other services listed in this document, Nextcloud requires an app password to be generated in the bot’s security settings to send webhooks. The app password will be used to generate the credential that needs to be made for the bot account. Credentials can be created in the **Authorization Manager** section on the **Outbound REST Requests** page in Web ClockWise. Click the **Add New Credential** button, give the credential a name, set Authorization Type to Basic, and enter the Username and Password for the account. Finally, save the credential:

## Outbound REST Requests

API Endpoints Authorization Manager

Credential Name \*

Nextcloud

Authorization Type \*

Basic

Username \*

.....

Password \*

.....

Save Credential

Cancel

When creating the API endpoint in the next step, make sure to select the credential you just created. You will also need to add a Header with the **Key** set to **Ocs-Apirequest** and the **value** set to **True**. This is a Nextcloud-specific header that must be present for the webhook to be successful:

**Credential \***

Nextcloud


**Http Method \***

POST

**Base URL \***

https://cloud.digidescorp.com/ocs/v2.php/apps/spreed/api/v1/chat/j8ogjkvf

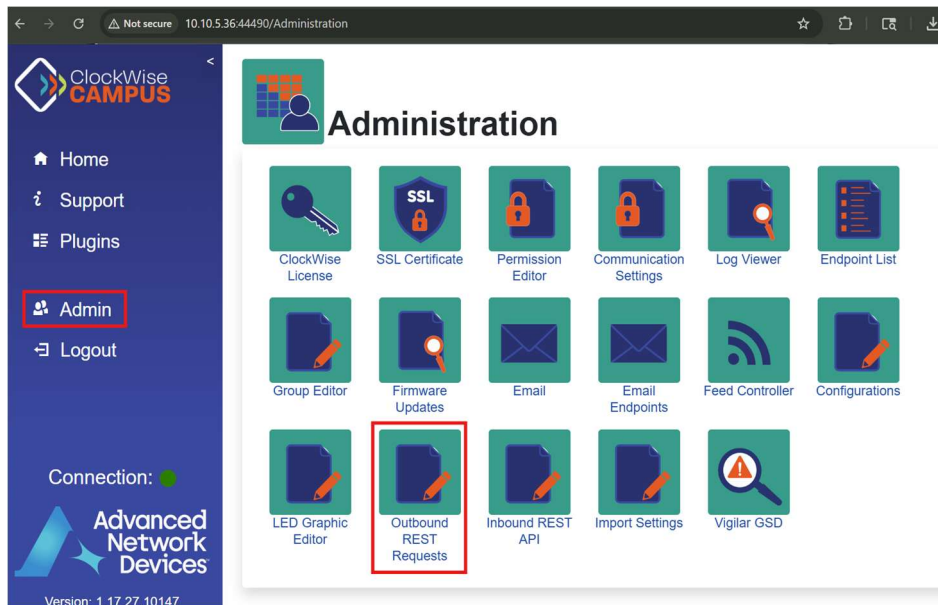
**Headers**

Key	Value	Actions
Ocs-Apirequest	True	

One final thing to note is that Nextcloud does not support the **{text}** parameter. You must use **{message}** in the Body Text section instead.

## CREATING THE API ENDPOINT IN CW WEB

You will need to log in to Web ClockWise as an administrator in order to create the API endpoint. Once logged in as an admin, navigate to the **Admin > Outbound REST Requests** page (/HttpAction):



Click the **Add New Endpoint** button to configure the API Endpoint. Give the API Endpoint a **Name**. Set the **HTTP method** to **POST**. Paste the URL for the webhook into the text box. Leave Body Type as **application/json**. Finally, enter a message of your choosing:

### Outbound REST Requests

API Endpoints    Authorization Manager

---

Endpoint Name \*

Credential \*

Http Method \*

Base URL \*

[Add Header](#)

[Add Query String Parameter](#)

Body Type \*

Body Text

For Zoom and Nextcloud, you must also add the headers specified in their specific setup instructions above. Nextcloud will also need the generated credential selected.

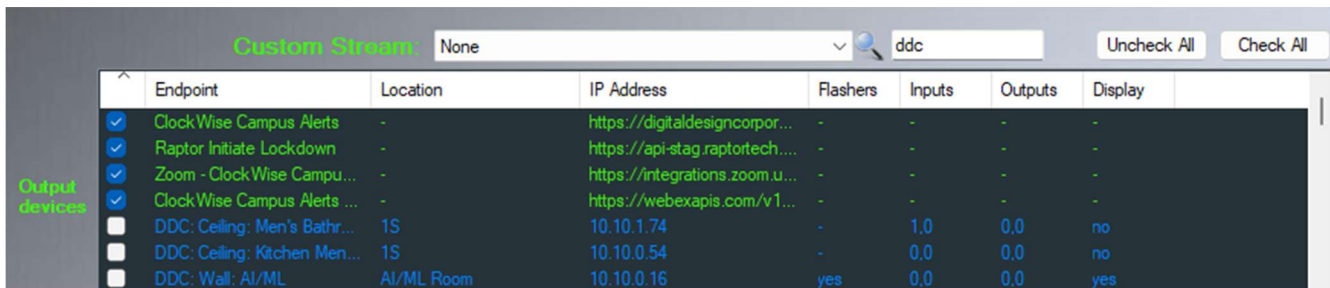
The message must be in JSON format. Each service has its own guides on how to properly formulate messages. However, for basic functionality, `{"text": "{Notification.Body}"}` can be used to send the text configured on the ClockWise message itself. This string works for all services included in this app note (except for Nextcloud which requires `{"message": "{Notification.Body}"}`). Additional information on how to formulate messages for each service can be found at the bottom of this document.

The endpoint should be tested to confirm functionality. A message will be sent when testing the request in Web ClockWise. The endpoint can be saved after confirming functionality.

## ADDING THE ENDPOINT AS A RECIPIENT OF A NOTIFICATION

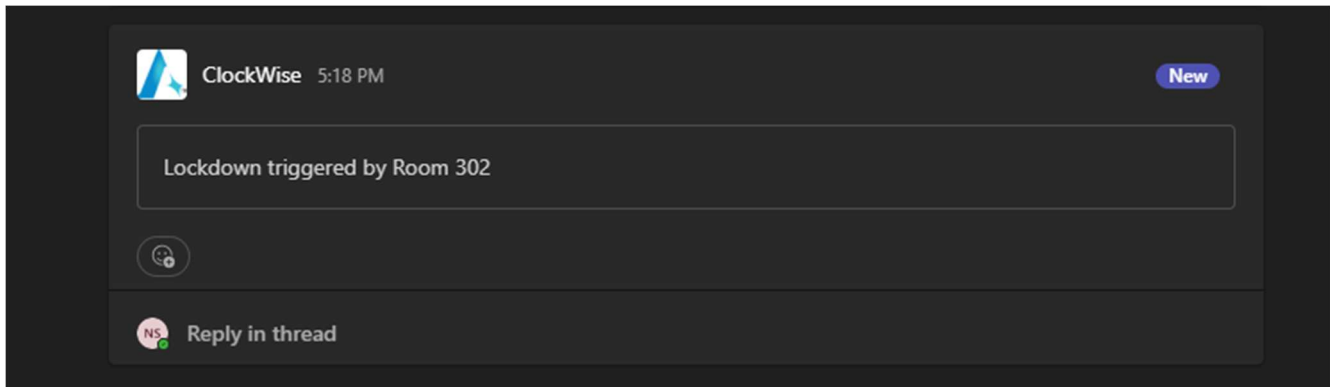
To add the endpoint as a recipient of the notification, you will first need to restart the ClockWise service. This can be done by right clicking the service LED in the top right corner of the application and selecting the Restart Service pop-up.

After the service successfully restarts, you can add the API Endpoint as an Output Device for a Notification:



Endpoint	Location	IP Address	Flashers	Inputs	Outputs	Display
<input checked="" type="checkbox"/> ClockWise Campus Alerts	-	https://digitaldesigncorpor...	-	-	-	-
<input checked="" type="checkbox"/> Raptor Initiate Lockdown	-	https://api-stag.raptortech...	-	-	-	-
<input checked="" type="checkbox"/> Zoom - ClockWise Campu...	-	https://integrations.zoom.u...	-	-	-	-
<input checked="" type="checkbox"/> ClockWise Campus Alerts ...	-	https://webexapis.com/v1...	-	-	-	-
<input type="checkbox"/> DDC: Ceiling: Men's Bathr...	1S	10.10.1.74	-	1.0	0.0	no
<input type="checkbox"/> DDC: Ceiling: Kitchen Men...	1S	10.10.0.54	-	0.0	0.0	no
<input type="checkbox"/> DDC: Wall: AI/ML	AI/ML Room	10.10.0.16	yes	0.0	0.0	yes

When this notification gets sent out, the chat service that is selected as a recipient will receive the notification as well. You can test this by triggering the alert and confirming the message is sent in the chat:



## FORMATTING

More information on formatting messages for each service can be found below:

Google: <https://developers.google.com/workspace/chat/api/reference/rest/v1/spaces.messages#Message>

Teams: <https://learn.microsoft.com/en-us/microsoftteams/platform/webhooks-and-connectors/how-to/add-incoming-webhook?tabs=newteams%2Cdotnet>

Webex: <https://developer.webex.com/messaging/docs/basics>

Zoom: <https://developers.zoom.us/docs/api/chat/?ampDeviceId=ed77dc54-5239-461e-b5cf-d7efd7f4ed7c&ampSessionId=1774542326261#tag/chat-messages/post/chat/users/{userId}/messages>

Nextcloud: [https://docs.nextcloud.com/server/stable/developer\\_manual/client\\_apis/OCS/ocs-api-overview.html](https://docs.nextcloud.com/server/stable/developer_manual/client_apis/OCS/ocs-api-overview.html)